

**SYSTEM AND METHOD FOR READ SYNCHRONIZATION  
OF MEMORY MODULES**

**TECHNICAL FIELD**

[001] The present invention relates to a processor-based system, and more particularly, to a processor-based system having a memory module with a memory hub coupling several memory devices to a processor or other memory access devices.

**BACKGROUND OF THE INVENTION**

[002] Processor-based systems, such as computer systems, use memory devices, such as dynamic random access memory ("DRAM") devices, to store instructions and data that are accessed by a processor. These memory devices are typically used as system memory in a computer system. In a typical computer system, the processor communicates with the system memory through a processor bus and a memory controller. The processor issues a memory request, which includes a memory command, such as a read command, and an address designating the location from which data or instructions are to be read. The memory controller uses the command and address to generate appropriate command signals as well as row and column addresses, which are applied to the system memory. In response to the commands and addresses, data is transferred between the system memory and the processor. The memory controller is often part of a system controller, which also includes bus bridge circuitry for coupling the processor bus to an expansion bus, such as a PCI bus.

[003] Although the operating speed of memory devices has continuously increased, this increase in operating speed has not kept pace with increases in the operating speed of processors. Even slower has been the increase in operating speed of memory controllers coupling processors to memory devices. The relatively slow speed of memory controllers and memory devices limits the data bandwidth between the processor and the memory devices.

[004] In addition to the limited bandwidth between processors and memory devices, the performance of computer systems is also limited by latency problems that increase the time required to read data from system memory devices. More specifically, when a

memory device read command is coupled to a system memory device, such as a synchronous DRAM (“SDRAM”) device, the read data are output from the SDRAM device only after a delay of several clock periods. Therefore, although SDRAM devices can synchronously output burst data at a high data rate, the delay in initially providing the data can significantly slow the operating speed of a computer system using such SDRAM devices.

[005] One approach to alleviating the memory latency problem is to use multiple memory devices coupled to the processor through a memory hub. In a memory hub architecture, a system controller or memory hub controller is coupled to several memory modules, each of which includes a memory hub coupled to several memory devices. The memory hub efficiently routes memory requests and responses between the controller and the memory devices. Computer systems employing this architecture can have a higher bandwidth because a processor can access one memory module while another memory module is responding to a prior memory access. For example, the processor can output write data to one of the memory modules in the system while another memory module in the system is preparing to provide read data to the processor. The operating efficiency of computer systems using a memory hub architecture can make it more practical to vastly increase data bandwidth of a memory system. A memory hub architecture can also provide greatly increased memory capacity in computer systems.

[006] Although there are advantages to utilizing a memory hub for accessing memory devices, the design of the hub memory system, and more generally, computer systems including such a memory hub architecture, becomes increasingly difficult. For example, in many hub based memory systems, the processor is coupled through a memory hub controller to each of several memory hubs via a high speed bus or link over which signals, such as command, address, or data signals, are transferred at a very high rate. The memory hubs are, in turn, coupled to several memory devices via buses that must also operate at a very high speed. However, as transfer rates increase, the time for which a signal represents valid information is decreasing. As commonly referenced by those ordinarily skilled in the art, the window or “eye” for when the signals are valid decreases at higher transfer rates.

With specific reference to data signals, the “data eye” decreases. As understood by one skilled in the art, the data eye for each of the data signals defines the actual duration that each signal is valid after various factors affecting the signal are considered, such as timing skew, voltage and current drive capability, and the like. In the case of timing skew of signals, it often arises from a variety of timing errors such as loading on the lines of the bus and the physical lengths of such lines.

[007] As data eyes of signals decrease at higher transfer rates, it is possible that one or more of a groups of signals provided by a memory device in parallel will have different arrival times at a memory hub to which the memory devices are coupled. As a result, not all of the signals will be simultaneously valid at the memory hub, thus preventing the memory hub from successfully capturing the signals. For example, where a plurality of signals are provided in parallel over a bus, the data eye of one or more of the particular signals do not overlap with the data eyes of the other signals. In this situation, the signals having non-overlapping data eyes are not valid at the same time as the rest of the signals, and consequently, cannot be successfully captured by the memory hub. Clearly, as those ordinarily skilled in the art will recognize, the previously described situation is unacceptable.

[008] One approach to alleviating timing problems in memory devices is to use a delay-locked loop (DLL) or delay line (DL) to lock or align the receipt of read data from a memory device to a capture strobe signal used to latch the read data in a memory hub. More specifically, a read strobe signal is output by the memory devices along with read data signals. At higher transfer rates, the timing of the read strobe signal can vary so that it cannot be reliably used to capture the read data signals in the memory hub. Further, even if the read data strobe could reliably capture the read data signals in the memory hub, the time at which the read data signals were captured could vary in relation to a core clock domain used to control the operation of the memory hub that is coupled to the memory device. In such case, the read data may not be present in the memory hub at the proper time. To alleviate this problem, the timing of the read data strobe signals is adjusted using the DLL or DL to generate a capture clock signal that can reliably capture the read data signals. The

DLL or DL is thus effective in preventing substantial drifting of a read data eye in relation to the core clock domain. As transfer rates increase, however, the timing specifications for the DLL or DL become more stringent and therefore increasingly difficult to meet. Furthermore, the amount of circuitry required to implement a suitable DLL or DL can materially reduce the amount of space that could otherwise be used for memory device circuitry, thereby either increasing the cost or reducing the storage capacity of such memory devices.

[009] There is accordingly a need for a system and method that avoids the need to precisely control the timing relationships between a memory hub clock domain and the receipt of read data signals at the memory hub in a manner that avoids the need for extensive DLL or DL circuitry.

#### SUMMARY OF THE INVENTION

[010] A memory module for a processor-based system includes a plurality of memory devices coupled to a memory hub. The memory hub includes a link interface for receiving memory requests for access to the memory devices and at least one memory device interface coupled to the memory devices. The memory device interface couples write memory requests and write data to the memory devices, and couples read memory requests to the memory device and read data from the memory device. The memory hub also includes a read synchronization module coupled to the memory device interface. The read synchronization module is operable to compare timing between the received read data and the memory requests. The memory hub further includes a memory sequencer coupled to the link interface and the memory device interface. The memory sequencer is operable to couple memory requests to the memory device interface responsive to memory requests received from the link interface, the memory sequencer further being operable to dynamically adjust operability responsive to the read synchronization module comparison.

### BRIEF DESCRIPTION OF THE DRAWINGS

- [011] Figure 1 is a block diagram of a computer system according to one example of the invention in which a memory hub is included in each of a plurality of memory modules.
- [012] Figure 2 is a block diagram of a memory hub used in the computer system of Figure 1, which contains read synchronization modules according to one example of the invention.
- [013] Figure 3 is a block diagram of one embodiment of a synchronization system according to one example of the invention.

### DETAILED DESCRIPTION OF THE INVENTION

- [014] Embodiments of the present invention are directed to a memory hub module having the capability to perform a read channel synchronization. Certain details are set forth below to provide a sufficient understanding of various embodiments of the invention. However, it will be clear to one skilled in the art that the invention may be practiced without these particular details. In other instances, well-known circuits, control signals, and timing protocols have not been shown in detail in order to avoid unnecessarily obscuring the invention.
- [015] A computer system 100 according to one example of the invention is shown in Figure 1. The computer system 100 includes a processor 104 for performing various computing functions, such as executing specific software to perform specific calculations or tasks. The processor 104 includes a processor bus 106 that normally includes an address bus, a control bus, and a data bus. The processor bus 106 is typically coupled to cache memory 108, which, as previously mentioned, is usually static random access memory ("SRAM"). Finally, the processor bus 106 is coupled to a system controller 110, which is also sometimes referred to as a "North Bridge" or "memory controller."
- [016] The system controller 110 serves as a communications path to the processor 104 for a variety of other components. More specifically, the system controller 110 includes a graphics port that is typically coupled to a graphics controller 112, which is, in turn, coupled to a video terminal 114. The system controller 110 is also coupled to one or more

input devices 118, such as a keyboard or a mouse, to allow an operator to interface with the computer system 100. Typically, the computer system 100 also includes one or more output devices 120, such as a printer, coupled to the processor 104 through the system controller 110. One or more data storage devices 124 are also typically coupled to the processor 104 through the system controller 110 to allow the processor 104 to store data or retrieve data from internal or external storage media (not shown). Examples of typical storage devices 124 include hard and floppy disks, tape cassettes, and compact disk read-only memories (CD-ROMs).

[017] The system controller 110 is coupled to several memory modules 130a,b...n, which serve as system memory for the computer system 100. The memory modules 130 are preferably coupled to the system controller 110 through respective high-speed links 134a and 134b, which may be optical or electrical communication paths or some other type of communications paths. The high speed link 134a is the downlink, carrying memory requests from the memory hub controller 132 to the memory modules 130a-n. The high speed link 134b is the uplink, carrying memory responses from the memory modules 130a-n to the memory hub controller 132. In the event the high-speed links 134a and 134b are implemented as optical communication paths, the optical communication paths may be in the form of one or more optical fibers, for example. In such case, the system controller 110 and the memory modules will include an optical input/output port or separate input and output ports coupled to the optical communication paths. The memory modules 130 are shown coupled to the system controller 110 in a multi-drop arrangement in which the high-speed links 134a and 134b are coupled to all of the memory modules 130. However, it will be understood that other topologies may also be used, such as a point-to-point coupling arrangement in which a separate high-speed link (not shown) is used to couple each of the memory modules 130 to the system controller 110. A switching topology may also be used in which the system controller 110 is selectively coupled to each of the memory modules 130 through a switch (not shown). Other topologies that may be used will be apparent to one skilled in the art.

[018] Each of the memory modules 130 includes a memory hub 140 for controlling access to 32 memory devices 148, which, in the example illustrated in Figure 1, are synchronous dynamic random access memory (“SDRAM”) devices. However, a fewer or greater number of memory devices 148 may be used, and memory devices other than SDRAM devices may, of course, also be used. In the example illustrated in Figure 1, the memory hubs 140 communicate over 4 independent memory channels 149 over the high-speed links 134a and 134b. In this example, although not shown in Figure 1, 4 memory hub controllers 128 are provided, each to receive data from one memory channel 149. A fewer or greater number of memory channels 149 may be used, however, in other examples. The memory hub 140 is coupled to each of the system memory devices 148 through a bus system 150, which normally includes a control bus, an address bus and a data bus.

[019] A memory hub 200 according to an embodiment of the present invention is shown in Figure 2. The memory hub 200 can be substituted for the memory hub 140 of Figure 1. The memory hub 200 is shown in Figure 2 as being coupled to four memory devices 240a-d, which, in the present example are conventional SDRAM devices. In an alternative embodiment, the memory hub 200 is coupled to four different banks of memory devices, rather than merely four different memory devices 240a-d, with each bank typically having a plurality of memory devices. However, for the purpose of providing an example, the present description will be with reference to the memory hub 200 coupled to the four memory devices 240a-d. It will be appreciated that the necessary modifications to the memory hub 200 to accommodate multiple banks of memory is within the knowledge of those ordinarily skilled in the art.

[020] Further included in the memory hub 200 are link interfaces 210a-d and 212a-d for coupling the memory module on which the memory hub 200 is located to a first high speed data link 220 and a second high speed data link 222, respectively. As previously discussed with respect to Figure 1, the high-speed data links 220, 222 can be implemented using an optical or electrical communication path or some other type of communication path. The link interfaces 210a-d, 212a-d are conventional, and include circuitry used for transferring data, command, and address information to and from the high speed data links

220, 222. As well known, such circuitry includes transmitter and receiver logic known in the art. It will be appreciated that those ordinarily skilled in the art have sufficient understanding to modify the link interfaces 210a-d, 212a-d to be used with specific types of communication paths, and that such modifications to the link interfaces 210a-d, 212a-d can be made without departing from the scope of the present invention. For example, in the event the high-speed data link 220, 222 is implemented using an optical communications path, the link interfaces 210a-d, 212a-d will include an optical input/output port that can convert optical signals coupled through the optical communications path into electrical signals.

[021] The link interfaces 210a-d, 212a-d are coupled to a switch 260 through a plurality of bus and signal lines, represented by busses 214. The busses 214 are conventional, and include a write data bus and a read data bus, although a single bi-directional data bus may alternatively be provided to couple data in both directions through the link interfaces 210a-d, 212a-d. It will be appreciated by those ordinarily skilled in the art that the busses 214 are provided by way of example, and that the busses 214 may include fewer or greater signal lines, such as further including a request line and a snoop line, which can be used for maintaining cache coherency.

[022] The link interfaces 210a-d, 212a-d include circuitry that allow the memory hub 200 to be connected in the system memory in a variety of configurations. For example, the multi-drop arrangement, as shown in Figure 1, can be implemented by coupling each memory module to the memory hub controller 128 through either the link interfaces 210a-d or 212a-d. Alternatively, a point-to-point, or daisy chain configuration can be implemented by coupling the memory modules in series. For example, the link interfaces 210a-d can be used to couple a first memory module and the link interfaces 212a-d can be used to couple a second memory module. The memory module coupled to a processor, or system controller, will be coupled thereto through one set of the link interfaces and further coupled to another memory module through the other set of link interfaces. In one embodiment of the present invention, the memory hub 200 of a memory module is coupled to the processor in a point-to-point arrangement in which there are no other devices coupled to the connection between



the processor 104 and the memory hub 200. This type of interconnection provides better signal coupling between the processor 104 and the memory hub 200 for several reasons, including relatively low capacitance, relatively few line discontinuities to reflect signals and relatively short signal paths.

[023] The switch 260 is further coupled to four memory interfaces 270a-d which are, in turn, coupled to the system memory devices 240a-d, respectively. By providing a separate and independent memory interface 270a-d for each system memory device 240a-d, respectively, the memory hub 200 avoids bus or memory bank conflicts that typically occur with single channel memory architectures. The switch 260 is coupled to each memory interface through a plurality of bus and signal lines, represented by busses 274. The busses 274 include a write data bus, a read data bus, and a request line. However, it will be understood that a single bi-directional data bus may alternatively be used instead of a separate write data bus and read data bus. Moreover, the busses 274 can include a greater or lesser number of signal lines than those previously described.

[024] In an embodiment of the present invention, each memory interface 270a-d is specially adapted to the system memory devices 240a-d to which it is coupled. More specifically, each memory interface 270a-d is specially adapted to provide and receive the specific signals received and generated, respectively, by the system memory device 240a-d to which it is coupled. Also, the memory interfaces 270a-d are capable of operating with system memory devices 240a-d operating at different clock frequencies. As a result, the memory interfaces 270a-d isolate the processor 104 from changes that may occur at the interface between the memory hub 230 and memory devices 240a-d coupled to the memory hub 200, and it provides a more controlled environment to which the memory devices 240a-d may interface.

[025] The switch 260 coupling the link interfaces 210a-d, 212a-d and the memory interfaces 270a-d can be any of a variety of conventional or hereinafter developed switches. For example, the switch 260 may be a cross-bar switch that can simultaneously couple link interfaces 210a-d, 212a-d and the memory interfaces 270a-d to each other in a variety of arrangements. The switch 260 can also be a set of multiplexers that do not provide the

same level of connectivity as a cross-bar switch but nevertheless can couple the some or all of the link interfaces 210a-d, 212a-d to each of the memory interfaces 270a-d. The switch 260 may also includes arbitration logic (not shown) to determine which memory accesses should receive priority over other memory accesses. Bus arbitration performing this function is well known to one skilled in the art.

[026] With further reference to Figure 2, each of the memory interfaces 270a-d includes a respective memory controller 280, a respective write buffer 282, and a respective cache memory unit 284. The memory controller 280 performs the same functions as a conventional memory controller by providing control, address and data signals to the system memory device 240a-d to which it is coupled and receiving data signals from the system memory device 240a-d to which it is coupled. The write buffer 282 and the cache memory unit 284 include the normal components of a buffer and cache memory, including a tag memory, a data memory, a comparator, and the like, as is well known in the art. The memory devices used in the write buffer 282 and the cache memory unit 284 may be either DRAM devices, static random access memory (“SRAM”) devices, other types of memory devices, or a combination of all three. Furthermore, any or all of these memory devices as well as the other components used in the cache memory unit 284 may be either embedded or stand-alone devices.

[027] The write buffer 282 in each memory interface 270a-d is used to store write requests while a read request is being serviced. In such a system, the processor 104 can issue a write request to a system memory device 240a-d even if the memory device to which the write request is directed is busy servicing a prior write or read request. Using this approach, memory requests can be serviced out of order since an earlier write request can be stored in the write buffer 282 while a subsequent read request is being serviced. The ability to buffer write requests to allow a read request to be serviced can greatly reduce memory read latency since read requests can be given first priority regardless of their chronological order. For example, a series of write requests interspersed with read requests can be stored in the write buffer 282 to allow the read requests to be serviced in a pipelined manner followed by servicing the stored write requests in a pipelined manner. As a result,

lengthy settling times between coupling write request to the memory devices 270a-d and subsequently coupling read request to the memory devices 270a-d for alternating write and read requests can be avoided.

[028] The use of the cache memory unit 284 in each memory interface 270a-d allows the processor 104 to receive data responsive to a read command directed to a respective system memory device 240a-d without waiting for the memory device 240a-d to provide such data in the event that the data was recently read from or written to that memory device 240a-d. The cache memory unit 284 thus reduces the read latency of the system memory devices 240a-d to maximize the memory bandwidth of the computer system. Similarly, the processor 104 can store write data in the cache memory unit 284 and then perform other functions while the memory controller 280 in the same memory interface 270a-d transfers the write data from the cache memory unit 284 to the system memory device 240a-d to which it is coupled.

[029] Further included in the memory hub 200 is a built in self-test (BIST) and diagnostic engine 290 coupled to the switch 260 through a diagnostic bus 292. The diagnostic engine 290 is further coupled to a maintenance bus 296, such as a System Management Bus (SMBus) or a maintenance bus according to the Joint Test Action Group (JTAG) and IEEE 1149.1 standards. Both the SMBus and JTAG standards are well known by those ordinarily skilled in the art. Generally, the maintenance bus 296 provides a user access to the diagnostic engine 290 in order to perform memory channel and link diagnostics. For example, the user can couple a separate PC host via the maintenance bus 296 to conduct diagnostic testing or monitor memory system operation. By using the maintenance bus 296 to access diagnostic test results, issues related to the use of test probes, as previously discussed, can be avoided. It will be appreciated that the maintenance bus 296 can be modified from conventional bus standards without departing from the scope of the present invention. It will be further appreciated that the diagnostic engine 290 should accommodate the standards of the maintenance bus 296, where such a standard maintenance bus is employed. For example, the diagnostic engine should have an

maintenance bus interface compliant with the JTAG bus standard where such a maintenance bus is used.

[030] Further included in the memory hub 200 is a DMA engine 286 coupled to the switch 260 through a bus 288. The DMA engine 286 enables the memory hub 200 to move blocks of data from one location in the system memory to another location in the system memory without intervention from the processor 104. The bus 288 includes a plurality of conventional bus lines and signal lines, such as address, control, data busses, and the like, for handling data transfers in the system memory. Conventional DMA operations well known by those ordinarily skilled in the art can be implemented by the DMA engine 286. The DMA engine 286 is able to read a link list in the system memory to execute the DMA memory operations without processor intervention, thus, freeing the processor 104 and the bandwidth limited system bus from executing the memory operations. The DMA engine 286 can also include circuitry to accommodate DMA operations on multiple channels, for example, for each of the system memory devices 240a-d. Such multiple channel DMA engines are well known in the art and can be implemented using conventional technologies.

[031] The diagnostic engine 290 and the DMA engine 286 are preferably embedded circuits in the memory hub 200. However, including separate a diagnostic engine and a separate DMA device coupled to the memory hub 200 is also within the scope of the present invention.

[032] Embodiments of the present invention provide a read synchronization module 297 for controlling the timing of read requests sent to the memory devices 240 so that read data signals are received at the memory hub 200 at the proper time in relation to a system clock signal used to establish a clock domain for the memory hub 200. Although a single synchronization module 297 is shown in Figure 2, it is to be understood that a plurality of synchronization modules 297 may also be used, for example, one per memory controller 280. Further, in the embodiment shown in Figure 2, the synchronization module 297 is shown in communication with the memory device 240c and the memory controller 280c. In some embodiments, the synchronization module 297 may be in communication with one or more memory devices and the controller 100 or memory hub 140 shown in Figure 1. As

mentioned above, the memory synchronization module 297 functions to synchronize the coupling of read data from the memory device with the core clock domain of the memory hub 200 as established by a system clock signal from the memory hub controller 128. Accordingly, if data is sent by the memory devices 148 either too early or too late, the read data might be coupled to the memory hub 200 at a time that is not synchronized to the core clock domain of the memory hub 200. Significantly, the synchronization module 297 allows the timing of a strobe signal used to capture read data signals to drift as needed so that the read data signals are captured at the proper time in relation to the core clock domain.

[033] Figure 3 illustrates a read synchronization module 300 according to an embodiment of the present invention that can be used as the read synchronization module 297 shown in Figure 2. It will be appreciated that Figure 3 is a functional block diagram representative of a suitable synchronization module and is not intended to limit the scope of the present invention. The functional blocks shown in Figure 3 are conventional, and can be implemented using well known techniques and circuitry. It will be further appreciated that control signals and other functional blocks have been omitted from Figure 3 in order to avoid unnecessarily obscuring the present invention, and that the description provided herein is sufficient to enable those ordinarily skilled in the art to practice the invention.

[034] Included in the read synchronization module 300 is a memory sequencer 304 that generates properly timed signals for controlling the operation of the memory devices 148 (Figure 1) or 240 (Figure 2). However, in alternative embodiments, the DMA engine 286 may be used for this purpose. The nature of the signals generated by the memory sequencer 304 will, of course, be determined by the nature of the signals used by the memory devices 148, 240. The timing of the signals controlling the operation of the memory devices 148, 240 control the time when read data signals are output from the memory devices 148, 240.

[035] A buffer 308 is used to store read data received from one or more of the memory devices 148, 240. The buffer 308 in Figure 3 is a first-in first-out (FIFO) buffer, such as a circular buffer, and may be implemented as known in the art. The buffer 308 is clocked

with a read strobe signal, which may also be referred to as a read clock signal. The read strobe signal is generated by the memory devices 148, 240 and is output from the memory devices 148, 240 along with read data signals. When the read data is clocked into the buffer 308 by the read strobe signal, *i.e.*, the read data are written to the buffer 308, a write pointer, 312 is incremented. The read data are clocked out of the buffer 308 and coupled to the memory hub controller 132 (Figure 1) by a core clock signal, which may be derived from a system clock signal. When data is clocked out of the buffer 308 by the core clock, *i.e.*, the read data are read from the buffer 308, a read pointer 314 is incremented. The read pointer 314 and the write pointer 312 are then compared by a comparator 316. Comparator 316 generates an adjust signal in response to the comparison. Generally, the relationship between the read pointer 314 and the write pointer 312 identifies the crossing margin from the memory device timing domain represented by the read strobe signal to the core clock timing domain – the “data eye”, as described above.

[036] The adjust signal is fed back to the memory sequencer 304. The data eye will decrease, *i.e.*, the read pointer 314 will be too close to the write pointer 312, if the read data are being coupled from the memory devices 148, 240 too early in relation to the core clock coupling the read data to the memory hub controller 128. In such case, the memory sequencer 304 reduces the rate at which read data are coupled from the memory devices 148. Conversely, the data eye will increase, *i.e.*, the read pointer 314 will be too far away from the write pointer 312, if the read data are being coupled from the memory devices 148 too late in relation to the core clock coupling the read data to the memory hub controller 128. In such case, the memory sequencer 304 increases the rate at which read data are coupled from the memory devices 148. As a result, the read data are coupled from the memory devices 148 at a rate that is adjusted to match the timing of the core clock signal.

[037] From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. Accordingly, the invention is not limited except as by the appended claims.